AKS素数判定アルゴリズムについての計算機を用いた実験的考察 (計算機科学とアルゴリズムの数理的基礎とその応用)

AUTHOR(S):

難波, 雄策; 神保, 秀司

CITATION:

難波, 雄策 ...[et al]. AKS素数判定アルゴリズムについての計算機を用いた実験的考察 (計算機科学とアルゴリズムの数理的基礎とその応用). 数理解析研究所講究録 2011, 1744: 189-192

ISSUE DATE:

2011-06

URL:

http://hdl.handle.net/2433/170952

数理解析研究所講究録
第1744巻 2011年 189-192

2010年度冬のLAシンポジウム [S5]

AKS素数判定アルゴリズムについての計算機を用いた実験的考察

Experimental Consideration on the AKS Algorithm with Computers

難波 雄策* 神保 秀司†
Yusaku NANBA Shuji JIMBO

1 はじめに

近年，情報通信技術の発達により高い安全性をもつ暗号システムが求められている．現在，そのための道具として素数判定アルゴリズムは，重要な役割を演じている．例えば，広く使われいている公開鍵暗号であるRSA暗号では，ランダムに選んだ同程度に巨大な2つの素数の積を鍵の一部として使っている．その際，必要とする大きさの範囲からランダムに選んだ整数に対して素数判定を施し素数と判定されたものを残すという方法が一般に使われる．

今世紀になって初めて素数判定問題を厳密に解く(確定的素数判定)決定性多項式時間アルゴリズムが公表され，開発した3人の研究者の頭文字をとってAKS素数判定アルゴリズムと呼ばれている [2][3]．しかしながら，原論文の公表後十分な改良が施されたものについてみても，前世紀に開発され広く使われているAPR法やECPP法などの確定的素数判定アルゴリズムと比べて実用的な範囲の入力に対する計算時間が極めて長いことが判明している．本論文では，AKS素数判定アルゴリズムをC言語を使って実装し，計算機上で動作させて得られた，主に計算効率についてのいくつかの実験結果について報告する．

*岡山大学大学院自然科学研究科
†第1著者に同じ

2 素数判定法について

次にあげるフェルマーの小定理は，素数判定アルゴリズムを設計するための基本原理の一つである．

定理 1 *(フェルマーの小定理)* n が素数なら，n の倍数でない整数 a に対して

$$a^{n-1} \equiv 1 \pmod{n} \tag{1}$$

適当な a を定めて式 (1) が成り立つか否かに従って p が素数であるか否かを判定する試験は，フェルマーテストと呼ばれる．

n を法とした剰余環 $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$ の要素で n と互いに素な要素 a $(\gcd(a,n) = 1)$ すべてからなる集合を $\mathbb{Z}_n^*$ で表したとき，$\mathbb{Z}_n^*$ の中に式 (1) を満たさない a が存在すれば，$\mathbb{Z}_n^*$ の要素のうちの半数以上の a が式 (1) を満たさないことを容易に示すことができる．しかしながら，すべての $a \in \mathbb{Z}_n^*$ について式 (1) が成り立つような1より大きい正整数 n が存在することが知られていて，そのような n は，カーマイケル数と呼ばれる．

フェルマーの小定理に加えて

$$p \text{が素数}, a \not\equiv \pm 1 \pmod{n} \Rightarrow a^2 \not\equiv 1 \pmod{n}$$

を考慮することにより Miller-Rabin 法が得られ，オイラーの規準を考慮することにより Solovay-Strassen 法が得られる．どちらも乱択アルゴリズム

であり，入力が素数であると判定したときの誤り確率が0にならないが，実行時間は大変短い．

一方，与えられた正整数 n が素数か合成数かを厳密に判別する確定的素数判定アルゴリズムとしては，APR法 [1] およびECPP法(楕円曲線素数証明法) [4] が挙げられる．ただし，APR法の実行時間は超多項式時間であり，ECPP法は乱択アルゴリズムである．

3 AKS素数判定法

3.1 準備

すべての整数からなる集合を $\mathbb{Z}$ で表し．すべての自然数 (正整数) からなる集合を $\mathbb{N}$ で表す．$\gcd(a,b)=1$ が成り立つとき a と b は互いに素であるという．本論文では特にことわらない限り対数の底は2とする．すなわち，$\log n = \log_2 a$ が成り立つ．

正整数 a と r が互いに素であるとき，$o_r(a)$ は $a^k \equiv 1 \pmod r$ を満たす最小の自然数 k を表し，r を法とした a の位数と呼ぶ．$\phi(r)$ は，r と互いに素な r 以下の正整数の個数を表す．

$t(n)$ が n の関数であるとき，$O^{\sim}(t(n))$ は $O(t(n)\cdot \mathrm{poly}(\log t(n)))$ の略記とする．例えば，任意の $\epsilon>0$ について $O^{\sim}(\log^k n) = O(\log^k n \cdot \mathrm{poly}(\log\log n)) = O(\log^{k+\epsilon} n)$ が成り立つ．

3.2 アルゴリズム

フェルマーの小定理を多項式についての合同式に拡張することにより次の定理が得られる．

定理 2 任意の素数 n および任意の正整数 a，r について，$\mathbb{Z}_n$ の要素を係数にもつ変数 X の1変数多項式間の ($\mathbb{Z}_n[X]$ に属する多項式間の) 合同式として次が成り立つ．

$$(X+a)^n \equiv X^n + a \pmod{X^r-1} \qquad (2)$$

この定理の中の合同式 (3) は，原論文 [3] では

$$(X+a)^n = X^n + a \pmod{X^r-1, n} \qquad (3)$$

と表されている．本論文でも，主にこちらの表記法を使う．

この定理は，フェルマーの小定理と同様に式 (3) を満たさない特定の r と a の組 (r,a) が n が合成数であることの証拠になっていることを保証している．さらに，r の値を条件

$$o_r(n) > \log^2 n \qquad (4)$$

を満たす整数に固定したとき，$1 \leqq a \leqq \lfloor\sqrt{\phi(r)}\log n\rfloor$ を満たすすべての整数 a について式 (3) が成り立てば，n が素数の累乗の形でなくてはならない [3]．

この原理に加えて，$\lfloor\sqrt{\phi(r)}\log n\rfloor < r$ が成り立ち，与えられた1より大きい整数が何らかの整数の2乗以上の累乗になっているか否かは，ニュートン法を使って $\log n$ の多項式時間で判定することができ，係数がすべて n 未満の非負整数である $r-1$ 次多項式で式 (3) の左辺を表すものが反復2乗法を使って $\log n$ の多項式時間で計算できることを考慮することにより，次に挙げるAKS素数判定アルゴリズムが得られる．

AKS素数判定アルゴリズム

入力: $1<n$ を満たす整数 n.

1. もし $n=a^b$，$b>1$ を満たす整数 a,b が存在すれば，「n は合成数」を出力して終了する．
2. $\gcd(r,n)=1$ および $o_r(n)>\log^2 n$ を満たす最小の正整数 r を見付ける．
3. もし $1<\gcd(a,n)<n$ を満たす正整数 a が $a \leqq r$ の範囲に存在すれば，「n は合成数」を出力して終了する．
4. もし $n \leqq r$ ならば，「n は素数」を出力して終了する．

5. $1 \leqq a \leqq \lfloor \sqrt{\phi(r)} \log n \rfloor$ の範囲の各整数 a について

$$(X+a)^n \neq X^n + a \pmod{X^r - 1, n}$$

を満たすか否かを検査し，もし満たすものが見付かれば「n は合成数」を出力して終了する．

6. 「素数」を出力して終了する．

上のアルゴリズムの時間計算量は，原論文 [3] では $O\tilde{}(\log^{7.5} n)$ であることが証明されている．さらに，素数の原始根についての Artin の予想などから，実際の実行時間が $O\tilde{}(\log^6 n)$ であることが強く予想されている．なお，Lenstra らにより，実行時間が $O\tilde{}(\log^6 n)$ であることを証明できるアルゴリズムの改良が提案されている [5]．

表 1: 10 進 50 桁までの $\alpha = 2$ での r

n の桁数	max	min	ave
5	347	179	254
10	1193	907	1049
20	4493	4001	4296
30	10667	9941	9737
40	17807	16811	17404
50	27751	26501	27255

表 2: 10 進 50 桁までの $\alpha = 1$ での r

n の桁数	max	min	ave
5	47	17	21
10	73	31	43
20	149	67	76
30	163	101	106
40	199	131	143
50	251	167	174

4 実験

AKS アルゴリズムを実装するにあたって，プログラムは C 言語で書き，多倍長整数を扱うことのできるライブラリである GMP[1] を使用した．また乱数発生には，環境ノイズをエントロピー源として利用し，乱数生成に活用する linux カーネルの /dev/random 及び /dev/urandom を使用した．

AKS アルゴリズムのステップ 2 における不等式の右辺 $\log^2 n$ の指数 2 を α に置き換え，α を本来の 2 から変化させて得られる r の値 r_α で表す．

第 1 の実験では，$\alpha, r_\alpha, \log^\alpha n$ の間の関係を調査する．α は，$\alpha \in \{0.5, 1, 1.5, 1.9, 2\}$ の範囲で変化させた．入力 n は，$2 \leqq n < 1000000$ の範囲で全数調査し，3 から 50 までの各整数についてその 10 進桁数をもつ一様乱数を 1000 回発生させて調査した．r_α の最大値，最小値などの数値を 10 進桁数別にまとめたものを表 1 および 2 に表す．

第 2 の実験では，2 つの大きな素数を掛け合わせて得られる合成数 n を入力した場合，ステップ 5 でどのような a が n が合成数であることの証拠になるかについて調査する．この実験においても，第 1 の実験と同様に r として各 $\alpha \in \{0.5, 1, 1.5, 1.9, 2\}$ に対する r_α を使った．$3 \leqq n < 10000$ の範囲の入力 n について全数調査した．さらに，3 から 50 までの各整数について一様分布に従って生成したその 10 進桁数をもつ 2 つの素数の積を n としてステップ 5 でどのような a が n が合成数であることの証拠になるかを調査するという工程を 100 回繰り返した．ただし，n が合成数であることの証拠となる a が存在しない場合，すなわち AKS アルゴリズムが誤動作したときは，その入力 n を出力させる．

入力が 3 つ素数の積である場合についても，素数の 10 進桁数が変化する範囲を 3 から 30 までとして同様の調査をした．

5 実験結果についてのまとめ

第 1 の実験では，r_2 の値は，$2 < n < 1000000$ を満たすすべて整数 n について高々 $\log^2 n$ の 2 倍未満であるという結果が得られた．α がより小さい場

[1] GNU MP home page, http://www.swox.com/gmp/.

合についても同様に $r_\alpha/\log^\alpha n$ の値が 10 を超えることはなかった．さらに，最大で 10 進 50 桁程度までのランダムに選んだ n について $r_\alpha/\log^\alpha n$ の値をみるとすべて 1 にかなり近い値になっていることが分かる．Artin の予想を考慮すれば，AKS アルゴリズムの実際の実行時間が $O^\sim(\log^6 n)$ であると大いに期待できる．

第 2 の実験では，実験に現れたすべてのステップ 5 において誤動作が発生することはなく，かつ，すべてのステップ 5 において $a=1$ が n が合成数であることの証拠であった．さらに，未確認の部分があるが，実験に現れたすべてのステップ 5 において $1 \leqq a \leqq r-1$ の範囲のすべての a が n が合成数であることの証拠になっているという結果も得られている．このことから，ステップ 2 の不等式の右辺を正定数 c を使って $c\log n$ の形に変更し，さらにステップ 5 で選ぶ a の個数を $O(\mathrm{poly}(\log\log n))$ になるように変更して，その結果得られるアルゴリズムが任意の合成数 n に対して正しく「n は合成数」と判定することを証明できれば，決定性 $O^\sim(\log^3 n)$ 時間確定的素数判定法が得られることになるが，10 進数十桁程度の大きさの n と r_α の組合せで，ステップ 5 で大部分の a が合同式を満たしてしまうものが極く稀に存在することは否定できない．なお，ステップ 2 で定まる r の値を小さくなるように AKS アルゴリズムを変更しても誤動作が生じ難いことは，瀬川らによる従来の実験結果 [6] からも強く示唆される．

6 今後の課題

初めに，今回の実験の後プログラムの不備が見付かり再実験により全般的な実験データの確認の必要性が判明していることを表明する．

今後の課題として，互いに素な合成数 n と整数 r の組 (n,r) に対してステップ 2 を除いて AKS アルゴリズムを適用したとき，ステップ 5 で多くの a が，例えば 1/10 以上の a が n が合成数であることの証拠にならないようなものを探索しそのようなものの中に r を法とした n の位数 $o_r(n)$ がどの程度大きいものが存在するかを調査する実験を実施して今回の結果を補強することを予定している．その場合，今回の実験で見送った高速フーリエ変換の原理を応用した多項式乗算の高速化を取り入れるなど実装の大幅な改良は必須と考える．また，入力として与える合成数については，今回使った同程度の大きさの 2 つの素数の積の形のもの以外に，多彩な素因数の構造をもつものを選ぶことが望ましい．さらに，フェルマーテストでの検出が困難な合成数であるカーマイケル数で巨大なものも実験の対象にする合成数として興味深い．

参考文献

[1] L. M. Adleman, C. Pomerance, and R. S. Rumely: On distinguishing prime numbers from composite numbers, *Annals of Mathematics*, 117, 173–206 (1983).

[2] M. Agrawal, N. Kayal, and N. Saxena: PRIMES is ib P, Preprint (`http://www.cse.iitk.ac.in/users/manindra/algebra/primality_original.pdf`) (2002).

[3] M. Agrawal, N. Kayal, and N. Saxena: PRIMES is ib P, *Annals of Mathematics*, 160, 781–793 (`http://www.cse.iitk.ac.in/users/manindra/algebra/primality_v6.pdf`) (2004).

[4] A. O. L. Atkin, and F. Morain: Elliptic Curves and Primality Proving, *Mathematics of Computation*, 61(203) 29–68 (1993).

[5] H. W. Lenstra, Jr. and C. Pomerance: Primality testing with Gaussian periods, Preprint (`http://www.math.dartmouth.edu/~carlp/PDF/complexity12.pdf`) (2005).

[6] 瀬川 紘・玉木 久夫: 合成数の AKS witness に関する実験的研究, **情報処理学会研究報告**, 2004(34) (2004-AL-94), 13–18 (2004).